

EXAMENUL DE BACALAUREAT – 2007
Proba scrisă la INFORMATICĂ
PROBA E, limbajul C/C++

Varianta 27

- ♦ Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- ♦ Timpul efectiv de lucru este de 3 ore.
- ♦ În programele cerute la subiectele II și III, identificatorii utilizați trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

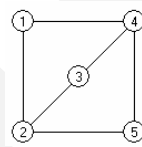
SUBIECTUL I (40 de puncte)

Pentru fiecare din itemii de la 1 la 8, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 5 puncte.

1. Se consideră că variabila `prim` memorează adresa de început a unei liste liniare simplu înlănțuite nevide. Orice element al listei memorează în câmpul `urm` adresa elementului următor din listă. Dacă expresia `prim->urm` este diferită de `NULL` și expresia `prim->urm->urm` are valoarea `NULL` atunci numărul de elemente din listă este egal cu :

a. 1 b. 0 c. 3 d. 2

2. Se consideră graful neorientat din figura alăturată. Numărul maxim de muchii ce pot fi eliminate din graf astfel încât graful parțial rezultat să fie conex este:



a. 0 b. 1 c. 2 d. 3

3. Se construiește un arbore în care nodul rădăcină memorează valoarea 20 iar fiecare nod neterminal are ca descendenți direcți noduri în care se păstrează **divizorii proprii** ai valorii din nodul părinte (numărul natural d este divizor propriu al numărului natural a , dacă d este divizor al numărului a și este diferit de 1 și de a). Câte noduri terminale (frunze) există în arbore ?

a. 5 b. 3 c. 10 d. 7

4. Fie a, b numere reale cu $a \leq b$. Numărul real x se găsește în afara intervalului închis $[a, b]$ dacă și numai dacă:

a. `x<=a || x>=b` b. `x<a || x>b` c. `x>=a && x<=b` d. `x<a && x>b`

5. Algoritmul alăturat atribuie variabilei `min` cea mai mică valoare întreagă dintre n numere întregi, mai mici decât 100, citite de la tastatură. Care dintre valorile de mai jos poate înlocui punctele de suspensie astfel încât algoritmul să furnizeze rezultatul corect?

```

citeste n (număr natural, n<50)
min←...
pentru i←1,n executa
|   citeste x (număr întreg x<100)
|   daca x<min atunci min←x
|
scrie min
    
```

a. 100 b. 1 c. 0 d. -100

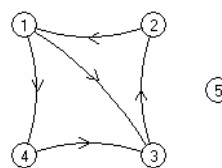
6. Câte caractere * se afișează în urma apelului `stea(3)`?

```

void stea (int x)
{int i;
  if(x>0){stea(x-1);
           for(i=1;i<=x;i++)
             printf("*"); |cout<<"*";
           }}
    
```

a. 6 b. 3 c. 9 d. 12

7. Considerând graful orientat din figura alăturată, stabiliți câte dintre vârful grafului au gradul extern (exterior) egal cu gradul intern (interior).



a. 2 b. 3 c. 1 d. 0

8. Se generează șiruri formate din caracterele 'A' și 'B'. Dacă se utilizează un algoritm backtracking care afișează în ordine, pentru $n=3$, șirurile **BBB**, **BBA**, **BAB**, **BAA**, **ABB**, **ABA**, **AAB**, **AAA** atunci pentru $n=4$, după șirul **ABAA** se va afișa șirul :
- a. **ABAB** b. **BABA** c. **AABA** d. **AABB**

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat în care s-a notat cu $x*y$ restul împărțirii întregi a lui x la y și cu $[x]$ partea întreagă a numărului real x .

```

citește n (număr natural, n>0)
repetă
| b ← n%10
| n ← [n/10]
| până când b >= n%10
scrie n

```

1. Ce se afișează pentru $n=23751$? **(5p.)**
2. Scrieți o valoare cu trei cifre care poate fi introdusă pentru variabila n astfel încât să se afișeze valoarea 0. **(3p.)**
3. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
4. Adăugați o structură alternativă la sfârșitul algoritmului care să afișeze, în plus, mesajul **DA** dacă cifrele numărului n sunt în ordine strict descrescătoare și mesajul **NU** în caz contrar. **(2p.)**

SUBIECTUL III (30 de puncte)

1. Scrieți programul C/C++ care, pentru un număr natural nenul n de cel mult 4 cifre, citit de la tastatură, afișează în ordine crescătoare, separate prin spațiu, primele n numere pare strict pozitive divizibile cu 5. De exemplu, pentru $n=6$ se afișează 10 20 30 40 50 60. **(10p.)**
2. Se consideră subprogramul **aparitii** care primește prin intermediul parametrului s un șir de maximum 100 de caractere iar prin intermediul parametrului x un caracter și returnează numărul de apariții ale caracterului x în șirul s .
 - a) Scrieți definiția completă a subprogramului **aparitii**. **(4p.)**
 - b) Scrieți programul C/C++ care citește de la tastatură două șiruri de maximum 100 de litere mici și verifică, utilizând apeluri ale funcției **aparitii** dacă cele două șiruri sunt anagrame (conțin aceleași litere, ordinea acestora fiind **diferită**). Se cere afișarea mesajului **anagrame** în caz afirmativ și a mesajului **nu sunt anagrame** în caz contrar. De exemplu, pentru șirurile **lada** și **dala** se afișează **anagrame**. **(6p.)**
3. Fișierul **bac.txt** conține pe primul rând un număr natural nenul n cu cel mult cinci cifre și pe fiecare dintre următoarele n linii câte două numere naturale a , b ($a < b$) cu cel mult 3 cifre fiecare, separate printr-un spațiu, numere ce reprezintă capetele unui interval închis. Se cere să se afișeze pe ecran, separate printr-un spațiu, două numere x și y ce reprezintă capetele intervalului de intersecție a celor n intervale date. Dacă intersecția lor este mulțimea vidă, se va afișa mesajul **multime vida**. De exemplu, dacă fișierul **bac.txt** conține:

4	se afișează
1 10	8 10
5 20	
8 12	
1 21	

 - a) Alegeți o metodă de rezolvare care să utilizeze eficient spațiul de memorie, descriind în limbaj natural metoda folosită și justificând eficiența acesteia (cel mult 6 rânduri) **(2p.)**
 - b) Scrieți programul C/C++ corespunzător metodei descrise la punctul a) **(8p.)**